**Step 1: Modify Controller to Return JSON Responses**

Update your UpdateRebuiltPartsAsync and CreateRebuiltPartsAsync methods to return a JSON response indicating success or failure.

**Example for UpdateRebuiltPartsAsync:**

csharp

Copy

[HttpPost]

[AllowAnonymous]

public async Task<IActionResult> UpdateRebuiltPartsAsync([FromBody] RebuiltPartsViewModel rebuiltPartsViewModel)

{

try

{

await UpdateRebuiltPartsAsync(rebuiltPartsViewModel); // Call your existing update method

return Json(new { success = true, message = "Rebuilt part updated successfully!" });

}

catch (Exception ex)

{

return Json(new { success = false, message = $"Error updating rebuilt part: {ex.Message}" });

}

}

**Example for CreateRebuiltPartsAsync:**

csharp

Copy

[HttpPost]

[AllowAnonymous]

public async Task<IActionResult> CreateRebuiltPartsAsync([FromBody] RebuiltPartsViewModel rebuiltPartsViewModel)

{

try

{

var rbMasterlistId = await CreateRebuiltPartsAsync(rebuiltPartsViewModel); // Call your existing create method

return Json(new { success = true, message = "Rebuilt part created successfully!", id = rbMasterlistId });

}

catch (Exception ex)

{

return Json(new { success = false, message = $"Error creating rebuilt part: {ex.Message}" });

}

}

**Step 2: Use Telerik Notification in the View**

In your Razor view, use the Telerik Notification component to display success or error messages based on the JSON response.

**Example Razor View:**

html

Copy

@using Telerik.UI;

<!-- Telerik Notification -->

@(Html.Kendo().Notification()

.Name("notification")

.Position(p => p.Top(50).Right(50))

.AutoHideAfter(5000) // Auto-hide after 5 seconds

)

<!-- Telerik Grid or other UI components -->

@(Html.Kendo().Grid<RebuiltPartsViewModel>()

.Name("grid")

.Columns(columns =>

{

columns.Bound(p => p.RbMasterlistId).Hidden();

columns.Bound(p => p.RebuiltStockNum);

columns.Bound(p => p.MmsStockCode);

columns.Bound(p => p.DetailedDesc);

columns.Command(command => { command.Edit(); command.Destroy(); }).Width(200);

})

.Editable(editable => editable.Mode(GridEditMode.PopUp))

.DataSource(dataSource => dataSource

.Ajax()

.Model(model => model.Id(p => p.RbMasterlistId))

.Read(read => read.Action("GetRebuiltPartsData", "YourController"))

.Update(update => update.Action("UpdateRebuiltPartsAsync", "YourController"))

.Create(create => create.Action("CreateRebuiltPartsAsync", "YourController"))

)

)

Run HTML

**Step 3: Handle the JSON Response in JavaScript**

Use JavaScript to handle the JSON response from the controller and show the appropriate message in the Telerik Notification.

**Example JavaScript:**

javascript

Copy

<script>

function showNotification(message, type) {

var notification = $("#notification").data("kendoNotification");

notification.show(message, type);

}

function onGridEditSuccess(e) {

if (e.response) {

if (e.response.success) {

showNotification(e.response.message, "success");

} else {

showNotification(e.response.message, "error");

}

}

}

function onGridCreateSuccess(e) {

if (e.response) {

if (e.response.success) {

showNotification(e.response.message, "success");

} else {

showNotification(e.response.message, "error");

}

}

}

$(document).ready(function () {

var grid = $("#grid").data("kendoGrid");

grid.bind("save", function (e) {

if (e.model.isNew()) {

// Handle create success

onGridCreateSuccess(e);

} else {

// Handle update success

onGridEditSuccess(e);

}

});

grid.bind("error", function (e) {

if (e.errors) {

var message = "An error occurred: " + e.errors;

showNotification(message, "error");

}

});

});

</script>